

**IN THE UNITED STATES DISTRICT COURT  
FOR THE SOUTHERN DISTRICT OF NEW YORK**

---

MICROSOFT CORPORATION,

Plaintiff,

v.

DATATERN, INC.,

Defendant.

---

CIVIL ACTION NO. 1:11-cv-02365-KBF  
CIVIL ACTION NO. 1:11-cv-02648-KBF

---

SAP AG and SAP AMERICA, INC.,

Plaintiff,

v.

DATATERN, INC.,

Defendant.

---

**DATATERN, INC.'S REPLY CLAIM CONSTRUCTION BRIEF**

Lee Carl Bromberg, BBO# 058480  
William A. Zucker, BBO #541240  
Erik Paul Belt, BBO# 558620  
McCarter English, LLP  
265 Franklin Street  
Boston, Massachusetts 02110  
Telephone: (617) 449-6500

and

245 Park Avenue  
New York, New York 10167  
(212) 609-6800

*Attorneys for Defendant  
DataTern, Inc.*

July 16, 2012

## TABLE OF CONTENTS

TABLE OF AUTHORITIES .....	ii
TABLE OF ABBREVIATIONS .....	iii
I. THE ‘402 PATENT .....	1
A. “LOGICAL TABLE” AND “NORMALIZED RELATIONAL SCHEMA OBJECT” .....	1
B. LOGICAL PRIMARY KEY COLUMN .....	3
C. “DESIGNATING” AND “GENERATING” .....	4
II. THE ‘502 PATENT .....	4
A. “OBJECT MODEL” .....	4
B. “TO CREATE AT LEAST ONE INTERFACE OBJECT” .....	6
C. “ASSOCIATED WITH AN OBJECT CORRESPONDING TO A CLASS ASSOCIATED WITH THE OBJECT ORIENTED SOFTWARE APPLICATION” .....	7
D. “RUNTIME ENGINE” .....	8
E. “UTILIZING A RUNTIME ENGINE WHICH INVOKES SAID AT LEAST ONE INTERFACE OBJECT . . . ” .....	9
CONCLUSION .....	10

# **TABLE OF AUTHORITIES**

	<b>Page(s)</b>
<b>CASES</b>	
<i>Environmental Designs, Ltd. v. Union Oil Co. of California</i> , 713 F.2d 693 (Fed. Cir. 1983).....	1
<i>Johnson Worldwide Assocs., Inc. v. Zebco Corp.</i> , 175 F.3d 985 (Fed. Cir. 1999).....	4
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	6
<i>Rexnord Corp. v. Laitram Corp.</i> , 274 F.3d 1336 (Fed. Cir. 2001).....	8
<i>SRI Intern. v. Matsushita Elec. Corp. of America</i> , 775 F.2d 1107 (Fed. Cir. 1985) (en banc).....	5
<i>Vitronics Corp. v. Conceptoronic, Inc.</i> , 90 F.3d 1576 (Fed. Cir. 1996).....	2

**TABLE OF ABBREVIATIONS**

DataTern	DataTern, Inc., defendant and counterclaim plaintiff
DT Ex.	DataTern Exhibit (previously filed with the Court)
DT Op. Br.	DataTern's Opening Claim Construction Brief, filed May 11, 2012
DT Resp. Br.	DataTern's Responsive Claim Construction Brief, filed July 2, 2012
Gupta Tr.	Transcript of the Deposition of Neeraj Gupta
JCCS	Joint Claim Construction Statement, filed May 11, 2012
Plaintiffs	Microsoft Corporation, SAP AG, and SAP America, Inc.
Pls. Ex.	Plaintiffs' Exhibit (previously filed with the Court)
Pls. Op. Br.	Plaintiffs' Opening Claim Construction Brief, filed May 11, 2012
Pls. Resp. Br.	Plaintiffs' Responsive Claim Construction Brief, filed July 2, 2012
'402 patent	U.S. Patent No. 5,937,402
'502 patent	U.S. Patent No. 6,101,502

## I. THE ‘402 PATENT

### A. “LOGICAL TABLE” AND “NORMALIZED RELATIONAL SCHEMA OBJECT”

At the core of Plaintiffs’ proposed constructions for “logical table” and “normalized relational schema object” (“NRSO”) is an effort to rewrite Claim 1 of the ‘402 patent so that it is limited to one embodiment shown in Figure 5, which, according to Plaintiffs, applies to denormalized relational tables alone and requires a normalization process. But this proposal ignores the plain wording of the claims and other teachings of the specification, which make clear that the use of logical tables is not limited to that one embodiment.

Claim 1, of course, recites no normalization process, while other claims (*e.g.*, Claim 2) expressly recite one. Likewise, Claim 1 applies to any “physical tables” in a “relational database,” broadly contemplating either normal form or denormalized tables, while other claims require “capturing one or more *denormalized* relational schema from said relational database.” *See, e.g.*, Claims 8, 16, 19. Because Claim 1 is expressly not limited to denormalized tables and does not require a normalization process, Claim 1 cannot be limited to one embodiment in Fig. 5. Plaintiffs cannot overcome the doctrine of claim differentiation, especially because, as detailed below, the specification itself disputes Plaintiffs’ narrowing construction.<sup>1</sup>

First, the patent teaches that “[a]ny captured physical table can be used to define logical tables.” ‘402 patent at 5:4-5 (emphasis added). The patent then explains that the “physical table may or may not have a primary key.” *Id.* at 5:5-6. In other words, the table may or may not be normalized. By having a primary key, each row is unique, thus rendering the table at least in first order normal form. *Id.* at 1:36; *see also* Hosking Decl. (Docket # 91) at ¶ 48.

---

<sup>1</sup> *See, e.g., Environmental Designs, Ltd. v. Union Oil Co. of California*, 713 F.2d 693, 699 (Fed. Cir. 1983) (“It is improper for courts to read into an independent claim a limitation expressly set forth in another claim”).

Indeed, the patent starts with the premise that *normalized* tables in a relational database may pose problems for object oriented applications and thus require a solution:

[N]ormalized tables are structured quite differently from objects. More particularly, the collection of information in the fields of the tables often fails to match the collection of information that would typically be found in a well designed object. *Consequently, such objects can cause problems for the software application.*

‘402 patent at 1:32-42 (emphasis added). That solution is the use of logical tables. *Id.* at 1:45-52 (“In accordance with the present invention logical tables and logical keys are employed to facilitate interaction between user applications and a relational database”). In other words, logical tables can be used with normal form and normalized tables, not just denormalized tables.

Plaintiffs, on the other hand, ignore these disclosures, arguing instead that the patent “discloses two embodiments, one claimed and another unclaimed.” Pls. Resp. Br. at p. 6. But a construction that excludes a preferred embodiment is “rarely if ever correct.” *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996). Plaintiffs focus only on a second disclosed embodiment, which tackles the problem of how to deal with denormalized tables by using a normalization process. But as seen above, a normalization process is not required for normal tables, either by the claim wording or the specification.<sup>2</sup>

Finally, Plaintiffs misquote the record. For example, contrary to Plaintiffs’ argument, DataTern’s expert, Mr. Gupta, did not state that logical tables must be created only from denormalized tables. The quote upon which Plaintiffs rely is taken out of context. The very next question and answer make it clear that Mr. Gupta is simply stating that a logical table can be created from a table in normal form *or* from a denormalized table after it has been normalized.

---

<sup>2</sup> The claims require only “defining” a logical table, not “creating” one. That means that the logical table is an abstraction represented by the NRSOs, and not, as Plaintiffs argue, a physical table. Indeed, Fig. 11 also shows that a logical table is an abstraction represented by the NRSOs. Claim 1 itself requires the formation of NRSO “representing the logical table,” which is precisely what Fig. 11 illustrates.

See Pls. Ex. GG, Gupta Tr. at 91:15-23. Similarly, Plaintiffs' reliance on the discussion of the Maloney prior art reference in the prosecution history omits critical context. Plaintiffs' quoted section is preceded by "Maloney teaches a system of logical schemas used to permit access and manipulation of a relational database *without knowledge of the structure of the database*" (emphasis added) and ends with "[t]he resulting logical tables in Maloney cannot, accordingly, be considered similar to the logical table represented by the normalized relational schema objects of the presently claimed invention." See Pls. Ex. T, 10/21/98 Response to Office Action, at p. 12. Thus, the distinction between Maloney and the '402 patent is that Maloney teaches a different kind of logical table altogether, not the use of a normalization process.

#### **B. LOGICAL PRIMARY KEY COLUMN**

Plaintiffs' definition is incomplete. A person of ordinary skill in the art would understand that the logical primary key column means that "each data row of the logical table is unique (and by extension uniquely identified by a logical primary key value)." DT Ex. B, Gupta Decl. at p. 9. In quoting the '402 patent (at 9:1-4), Plaintiffs once again omit key text from the specification. The full quote reads:

In the normalized form, the identity of an object in the relational table is provided and supported by a primary key constraint that is provided and supported by the relational database. The primary key constraint specifies a set of one or more columns of a table and imposes that the designated columns can contain values whose combination must be unique in the table.

'402 patent at 8:65-9:4. Thus, the primary key (*i.e.*, the combination) need not be contained in any one column but rather need only be associated with one or more columns. The logical primary key constraint or value can specify a set of one or more columns; thus, the logical primary key value is associated with a column of a logical table and the column of a logical table is associated with the logical primary key value.

### C. “DESIGNATING” AND “GENERATING”

The claims themselves do not limit the terms “designating” and “generating” and thus carry the heavy presumption that the terms have their plain, unmodified meaning. *Johnson Worldwide Assocs., Inc. v. Zebco Corp.*, 175 F.3d 985, 989 (Fed. Cir. 1999). Further, the specification teaches modes allowing user input and automatic modes without. Compare ‘402 patent at 5:4-14 (disclosing a user input mode) with 3:66 – 4:8 (automated mode). Further, contrary to Plaintiffs’ argument, the patentee did not disclaim a mode in which “creating” excludes user input. In the passage from the prosecution history on which Plaintiffs rely, the patentee distinguished Chang because the features of Chang “do not disclose, teach or suggest generating object classes, associated with a normalized relational schema object *responsive to the normalized relational schema object*, as required by claim 1.” Pls. Ex. II, 12/06/11 Response to Non-Final Office Action, at 14 (emphasis in original). In other words, Chang was not distinguished based on excluding user input. One cannot, therefore, read there to be any disclaimer as a result of the reference to Chang.

## II. THE ‘502 PATENT

### A. “OBJECT MODEL”

An “object model” is a “template with a predetermined standardized structure that may include representing object classes, attributes of those object classes, and inheritance relationships among classes.” This definition is taken directly from the specification. *See* ‘502 patent at 2:39-44. Ignoring the claims and the specification, Plaintiffs seek to add the limitations that the “object model” must include (1) “a set of classes and the relationships among those classes” that (2) “under[lie] the implementation of the object oriented software application.”

Via the first part of the proposed definition, Plaintiffs want this Court to adopt a new “object model class” limitation that is found nowhere in the claims or specification and based



solely on their desire to distinguish the accused BusinessObjects product. *See* Pls. Resp. Br. at 17 (referring to “SAP’s accused BusinessObjects software” in the context of the “object model” claim construction). Claim construction, however, must be accomplished without reference to the accused products. *SRI Intern. v. Matsushita Elec. Corp. of America*, 775 F.2d 1107, 1118 (Fed. Cir. 1985) (en banc). Further, the object model need only be a template with a predetermined standardized structure. The exemplary object model 14 described at 2:39-44, for example, may not include any classes because it specifies no behavior, as required by the agreed definition of a class.<sup>3</sup> *See also* Pls. Ex. GG, Gupta Tr. at 209:11-18 (“There’s no teaching that the object model must include behaviors that are not used in the mapping process”).

The second part of the proposed definition assumes, in effect, that the user is writing an object oriented application based on an object model. Plaintiffs, however, fail to recognize that the ‘502 patent is directed to facilitating access between an object oriented application (which may already exist) and a relational database, not the creation of the object oriented application itself. The claims, for example, do not recite the creation of an object oriented application. Further, there is nothing in the ‘502 patent or its file history that suggests that the object model must predate the object oriented application, as Plaintiffs’ proposed construction would require. *See, e.g.*, Hosking Decl. at ¶ 32 (acknowledging that an object model can be created from an object oriented application, and therefore need not predate the application). Moreover, the specification expressly states that the object model can be created from the database schema and thus need not come from or underlie the object oriented application. ‘502 patent at 1:56-58.

---

<sup>3</sup> Plaintiffs agreed that, in the context of the ‘502 patent claims, “a class” is “a definition that specifies attributes **and behavior** of objects, and from which objects can be instantiated.” JCCS at 2 (emphasis added).

**B. “TO CREATE AT LEAST ONE INTERFACE OBJECT”**

Claim 1 requires, simply, “employing the map to create at least one interface object . . . .” Plaintiffs, however, would like to muddy the existing claim term by adding yet another new limitation in which the creation of the interface object requires an unclaimed, unrecited two-step process involving (1) generating code for a new interface object class before (2) instantiating the interface object from the new interface object class. The claims themselves belie this construction. Nothing in Claim 1 states how or when the map is used to create an interface object. Moreover, while Claim 1 does not require code generation, Claim 10 expressly uses a “code generator.” Putting aside the fact that Claim 1 is directed to a method and Claim 10 to a product, and contrary to Plaintiffs’ argument, these claims by law must be deemed to have different scopes. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1324-25 (Fed. Cir. 2005) (difference between two independent claims, one which included a particular limitation and one which did not, showed that the limitation could not be read into the other claim).

Other claims, such as 21, require “creating an *instance* of a runtime engine *class* by the runtime engine for every interface object *instance*.” In other words, the patentee certainly knew how to distinguish between instances and classes but did not do so when claiming an “interface object.” Although the patentee chose not to expressly claim the creation of a class, Plaintiffs, in effect, want this Court to decipher a new term, “interface object *class*,” that appears nowhere in the claims or specification.<sup>4</sup>

Plaintiff’s suggestion that Fig. 1 and the related discussion is the only disclosure of the creation of an interface object is simply wrong. For example, the specification (at 6:31-32)

---

<sup>4</sup> As with “object model class,” Plaintiffs want this Court to adopt “interface object *class*” based on the improper reference to their accused products. *See* Pls. Resp. Br. at 25 (referring to “SAP’s accused BusinessObjects products” and “Microsoft’s accused software” in the context of the “creat[ing] at least one interface object” dispute).

discloses that “FIG. 7 illustrates the sequence of actions that take place when a business object<sup>5</sup> *creates* a Dsl object”—which Plaintiffs contend is an interface object. *See* Pls. Resp. Br. at 34. No “code generator” or “generat[ion of] code for [an interface object] class” is mentioned with respect to Fig. 7.

Finally, Plaintiffs would also have this Court add the additional limitation, not found in the claims, that the interface object “not [be] part of or generated by the object-oriented application.” The claim wording does not require a particular element to perform or not perform the creation step, and neither should this Court, despite Plaintiffs’ invitation to do so. The disputed phrase is part of a larger phrase that requires “employing the map to create at least one interface object associated with an object corresponding to a class associated with the object oriented application.” In other words, the plain wording of the claim requires only that the interface object be associated with an object of the object oriented application and does not care who or what generates the interface object.

**C. “ASSOCIATED WITH AN OBJECT CORRESPONDING TO A CLASS ASSOCIATED WITH THE OBJECT ORIENTED SOFTWARE APPLICATION”**

By its plain wording, this phrase requires three relationships between four elements—an “interface object,” an “object,” a “class,” and an “object oriented software application.” Plaintiffs would like to construe the phrase, however, to require relationships between different and newly-concocted elements—an “interface object *class*” and “object model *class*.” Of course, the claims do not include these “class” limitations, and neither should this Court.

Plaintiffs agree that “an object corresponding to a class”—which is a portion of the larger disputed phrase—means “an object instantiated from a class.” *See* Pls. Resp. Br. at 33.

---

<sup>5</sup> Fig. 5 shows that Business Objects 58 are part of the object oriented End User Application 22.

Plaintiffs further agree that the terms “an object” and “a class” have different meanings in the context of the disputed phrase. See JCCS at 2 (identifying agreed upon constructions for “a class” and “an object”).

Ignoring this previously agreed distinction between an object and a class, however, Plaintiffs propose that the recited association between objects be construed as an association between *classes*. See Pls. Resp. Br. at 33. Of course, the claim wording itself contradicts this proposal. Claim 1 recites creating an “interface *object* associated with an *object*.” The specification also refers to an association between objects, not classes. See ‘502 patent at 3:2-7 (“When an *object* has *associations* to *other objects*, an attribute in the object points to one or multiple other object”) (emphasis added). The association, discounted by Plaintiffs, requires no code generator. The Court should reject Plaintiffs attempt to limit the claims to one of multiple disclosed embodiments. *Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1334 (Fed. Cir. 2001) (reversing district court for improperly limiting the claim to one disclosed embodiment).

Finally, a “class associated with the object oriented software application” requires no interpretation. See, e.g., Hoskings Decl. ¶¶ 17, 19 (describing an object oriented application as a collection of classes); Pls. Resp. Br. at 18 (referring to classes as a touchstone of object oriented applications). But, in any event, a “class associated with the object oriented software application” is not an “object model class”—at least because an “object model” is different from an “object oriented software application.”

#### **D. “RUNTIME ENGINE”**

The runtime engine is simply a service that the object oriented application uses to access the relational database. Plaintiffs’ efforts to straightjacket this term with further narrowing limitations is unsupported by the claims and other intrinsic evidence, which shows that the runtime engine need not directly use the map, and need not be sharply distinguished from the

application. DT Resp. Br. at 34-37. Contrary to Plaintiffs' characterization, the USPTO examiner did not define "runtime engine." Rather, the examiner allowed the claims because the "runtime engine" was properly distinguished from "runtime environment" and "runtime library." *See* Pls. Ex. O, Notice of Intent to Issue Reexamination Certificate, at DTM4 001503. The examiner also found that "[t]he Eisner Affidavit is supported by the cited PCMag.com reference *distinguishing* the phrases 'runtime engine' from 'runtime environment' and 'runtime library.'" *Id.* (emphasis added); *see also id.* at DTM4 001524 (circling "runtime environment" and "runtime library" in the list of terms related to the definition of "runtime engine"). DataTern also stressed "that the basis for patentability of each of these claims is based on the *totality* of the particular features recited therein, *even absent the Eisner Affidavit*." Pls. Ex. Q, Comments on Reasons for Patentability and/or Confirmation, at DTM4 001476 (emphasis added).

While the difference between a "runtime engine" and "runtime environment" may be subtle to a lay person, in the context of the claims, DataTern's construction captures the difference. In contrast, Plaintiffs seek to create bright line distinctions based on extrinsic evidenced when, in reality, their extrinsic evidence does not support the alleged distinctions. *See* Pls. Ex. P, Printout of webpage from PCMag.com (explaining that, although it "' 'is the essential runtime engine,' . . . an operating system is not classified as a runtime engine, but . . . would fall into the 'runtime environment' category.>").

**E. "UTILIZING A RUNTIME ENGINE WHICH INVOKES SAID AT LEAST ONE INTERFACE OBJECT . . ."**

In the phrase "utilizing a runtime engine which invokes said at least one interface object with the object oriented application to access data from the relational database"—the word "utilizing" has its ordinary meaning, "runtime engine" has the foregoing definition, and

“invokes” means “calls.” JCCS at 2 (defining “invokes” by agreement). To the extent any further definition is necessary, the phrase means “an object oriented application uses a runtime engine, which invokes an interface object, to access the relational database.”

Plaintiffs again propose to import a new requirement into the claims—namely, that the runtime engine accesses data “without using an intermediate, user modifiable mapping language.” In its original brief, DataTern explained that Plaintiffs’ proposed new requirement should not be imported into the claims because there was no clear and unmistakable disclaimer of the use of an intermediate, user modifiable mapping language. DT Op. Br. at 34-37.

### CONCLUSION

Dated: July 16, 2012

McCARTER & ENGLISH, LLP

/s/Lee Carl Bromberg  
Lee Carl Bromberg, Esq.  
Erik Paul Belt, Esq.  
William A. Zucker, Esq.  
McCarter & English, LLP  
265 Franklin Street  
Boston, MA 02110  
(617) 449-6500  
lbromberg@mccarter.com  
Erik Paul Belt, Esq.  
William A. Zucker, Esq.

and

245 Park Avenue  
New York, New York 10167  
(212) 609-6800

*Attorneys for Defendant  
DataTern, Inc.*

**CERTIFICATE OF SERVICE**

I certify that, on July 16, 2012, I served copies of DataTern, Inc.'s Responsive Claim Construction Brief upon counsel of record for the other parties in the above-referenced consolidated matters via ECF, with courtesy copies served via electronic mail.

/s/ Lee Carl Bromberg  
Lee Carl Bromberg